International Scientific Journal of Engineering and Management ISSN: 2583-6129
Volume: 01 Issue: 03 | Oct – 2022 DOI: 10.55041/ISJEM00113
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

# DESIGNING A SECURE N-TIER ARCHITECTURE FOR CLOUD ENVIRONMENTS

Ramasankar Molleti, Independent Researcher, Email: sankar276@gmail.com , USA

**ABSTRACT**

The study of building secure N-Tier architectures in cloud environments is crucial to the issue in today's distributed systems. It looks at the development of N-Tier architectures in cloud computing and outlines the main security components of each tier. The study provides detailed solutions for addressing the main security threats such as data privacy, access control, and threat identification. It also covers aspects such as design and implementation, emphasizing depth. The study discusses future works and directions which include zero-trust security systems, and AI security in the future. Thus, this research can be useful for organizations to increase the security of cloud-based N-Tier applications in the constantly developing threats.

**Keywords**: ***Cybersecurity, Data Privacy, Access Management, Threat Detection, Cloud Computing, Compliance, DevSecOps.***

## I. Introduction

Cloud computing shapes the organizations' IT environment, architecture, and governance. The focus of companies' key processes is shifting to various clouds, and the security of these N-Tier applications is a significant concern.

This study seeks to solve the significant problem of deploying sound security in all the layers of cloud N-Tier applications. This also discusses the N-Tier architectures from the perspective of cloud computing and how the previously used structures have been modified to fit in the cloud computing environment.

Basic features of a safe N-Tier cloud model are revealed and the major emphasis is made on protecting every tier starting from the presentation tier which interacts with users, through the application tier which handles business calculations up to the data tier which stores and controls valuable information.

These challenges extend from data privacy protection and compliance with the regulatory environment which is becoming stricter with each passing year, and the management of complex identity and access control systems in the distributed environment. This also presents detailed and possible solutions to each of these challenges to address them effectively in line with today's developments in the field.

Based on case studies and performance evaluation, it provides information about the effectiveness of security measures.

This study investigates new trends and possible future work that will define the next generation of secure cloud systems. This helps to anticipate the characteristics and development of new threats and alterations in the technology line.

## II. Evolution of N-Tier Architectures in Cloud Computing

The improvements that cloud computing has brought to the application structures, especially by the use of N-Tier architectures is a clear indication of a new age in application management.

### Traditional N-Tier Models

N-tier architecture is one of the main and most widely recognized principles of Software Design [1]. It traditionally consists of three primary tiers:

International Scientific Journal of Engineering and Management
ISSN: 2583-6129
Volume: 01 Issue: 03 | Oct – 2022
DOI: 10.55041/ISJEM00113
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

**Presentation Tier:** Manages the graphical user interface, or the user's interface with the system.
**Application Tier:** Contains business rules for the application and refers to the business processes.
**Data Tier:** Responsible for the gathering and returning of data.
There was an improvement in scalability, maintainability, and security coming from this fairly simple separation of concerns. N-tier models, especially the traditional ones, were often closely coupled to the physical resources and at the same time, were plagued by the problems of deployment.

**Adaptation to Cloud Environments**

The beginning of cloud computing in the mid-2000s brought about significant changes to N-Tier architectures:

**a) Virtualization:**

Cloud platforms added virtualization to the environment, which let the tiers be deployed on the virtual machines (VMs). This gave more flexibility and utilization of available resources.

**b) Elasticity and Auto-scaling:**

Cloud environments made it possible to provide or scale resources dynamically.

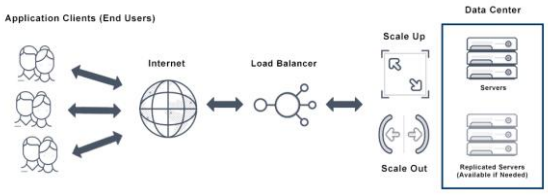


**Figure 1: Auto Scaling**

(Source: https://avinetworks.com/)

At this stage, every tier could adjust or expand as the need of the business arises, which is highly advantageous in accommodating a fluctuating workload [2].

**c) Platform as a Service (PaaS):**

PaaS offers simplification of the application tiers through deployment and management. For developers, there was a possibility to concentrate on code while not having to consider the infrastructure under them.

**d) Microservices Integration:**

By 2015, many organizations started decomposing monolithic, application tiers into microservices for convenience. This gave a chance for better scaling and easier modification of scaling.

**e) Containerization:**

New concepts such as containerization like Docker were introduced in 2013 and brought further changes to N-Tier deployment. Containers ensured some kind of homogeneity no matter the environment that was being used and made efficient management of resources.

**f) Serverless Computing:**

With the introduction of serverless computing by the major cloud providers around 2014-2015, the management of servers was taken to more extremes [3]. The model could be used on some aspects of N-Tier architectures mainly on the application tiers.

**g) Database as a Service (DBaaS):**

Cloud providers started providing managed services for the database tier, which means that the data tier management and scaling were made easier.

| Era | Key Development | Impact on N-Tier Architecture |
|---|---|---|
| Pre-2006 | Traditional N-Tier | Physical hardware constraints |
| 2006-2010 | Early Cloud Adoption | Virtualization of tiers |
| 2011-2015 | PaaS and Auto-scaling | Dynamic resource allocation |
| 2015-2020 | Microservices & Containers | Granular scaling, |

| | | |
|---|---|---|
| | | improved portability |
| 2020 onwards | Serverless & Edge Computing | Event-driven architectures, distributed processing |

**Table 1: Evolution of N-Tier Architecture in Cloud**

Due to the development of N-Tier architectures in cloud computing, the applications have become more flexible, scalable, and resilient. It has also brought new complexities, especially in the area of security and data, which must be solved in today's cloud-native N-Tier solutions.

## III. Core Components of Secure N-Tier Cloud Architecture

The security measures for each tier as well as the system have to be observed [4]. This study describes the basic elements and security challenges of each level of the N-Tier in the cloud environment.

### Presentation Tier Security

The presentation tier has to be protected from user-side attacks and provide a means of secure communication.

Key security components include:

**a) SSL/TLS Encryption:** Integrate HTTPS using the strong cipher suites to ensure data being transmitted over networks are secure.

**b) Content Security Policy (CSP):** Reduce the likelihood of Cross-Site Scripting (XSS) and other injection-based vulnerabilities.

**c) Web Application Firewall (WAF):** Use a WAF to filter and scrutinize HTTP traffic from between web apps and the Internet.

**d) Input Validation:** Employ strong measures in input validation on the client's end and on the server end to prevent injection attacks.

**e) Authentication and Authorization:** Use safe authentication mechanisms (for example OAuth 2. 0, OpenID Connect) and correctly check authorization [5].

### Application Tier Security

The application tier contains business logic, and it is quite vulnerable compared to the other tiers; thus, it needs strong security management against all the threats that may occur.

Key security components include:

**a) Microservices Security:** If microservices are in use, use a service mesh solution (Istio) for secure communication between services.

**b) API Security:** API gateways should be used for API management, monitoring, and even security. All the API endpoints should have rate limiting and the correct kinds of authentication.

**c) Containerization Security:** If using containers, use the Container Security Scanning Tools, and ensure that the container is configured with the principle of least privilege.

**d) Secure Coding Practices:** Fortify the code with secure coding norms followed by constant code review, and utilize SAST/DAST tools.

**e) Secrets Management:** The information should be stored and managed using cloud-native secrets management services (e.g., AWS Secrets Manager, Azure Key Vault, etc.) [6].

### Data Tier Security

The data tier is charged with the responsibility of managing and storing information and thus if an attack is to be launched then this is where it would be done.

Key security components include:

**a) Encryption at Rest:** Implement a high level of data protection standards (AES-256) to encrypt data managed in databases as well as object storage.

**b) Database Access Control:** It is suggested to enforce detailed access control measures and apply the principle of least privilege to the users of the database.

**c) Data Masking and Tokenization:** Use tokenization for payment card data.

International Scientific Journal of Engineering and Management
ISSN: 2583-6129
Volume: 01 Issue: 03 | Oct – 2022
DOI: 10.55041/ISJEM00113
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

**d) Audit Logging:** Ensure that audit logs of all the data accessed and changed are kept to the highest level of detail.

**e) Backup and Recovery:** Backup must be conducted on a routine basis and encrypted and the recovery processes should also be practiced frequently.

**Network and Communication Security**

Securing network communications between tiers and with external entities is crucial for maintaining the overall security of the N-Tier architecture [7].

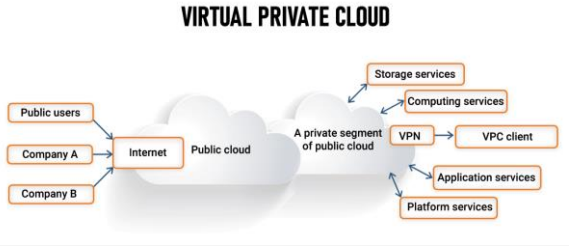


**Figure 2: Virtual Private Cloud**

(Source: https://images.spiceworks.com/)

Key security components include:

**a) Virtual Private Cloud (VPC):** Use VPC to organize resources and manage traffic.

**b) Network Segmentation:** Implement subnets and network access control lists (NACLs) to partition the tiers.

**c) Firewalls and Security Groups:** The firewalls and security groups should be enabled natively in the cloud.

**d) VPN and Direct Connect:** For hybrid cloud environments, use VPN or direct connect services for communication with the on-premises systems.

**e) DDoS Protection:** Use DDoS protection services offered by the cloud vendors or do it using other solutions.

When all of these centralized core security components are applied at every tier within the N-Tier architecture, organizations' cloud environment security increases greatly [8]. However, one has to remember that security is a continuous process and should be reviewed, updated, and changed according to new threats and risks.

## IV. Security Challenges and Mitigation Strategies

The use of N-Tier architectures in cloud settings brings about new security issues that must be met to provide the data and services' confidentiality, integrity, and accessibility.


**Figure 3: GDPR**

(Source: https://images.theconversation.com/)

**Data Privacy and Compliance**

Challenge:

Ensuring data privacy and maintaining compliance with regulations like GDPR, HIPAA, and PCI-DSS in a cloud environment can be complex due to the distributed nature of data storage and processing.

Mitigation Strategies:

**a) Data Classification:** Implement a robust data classification system to identify and appropriately protect sensitive information [9].

**b) Data Residency:** Use region-specific cloud services to ensure data remains within required geographical boundaries.

**c) Encryption:** Employ strong encryption for data at rest and in transit. Use the services that the cloud provider offers for managing keys.

**d) Privacy by Design:** Privacy should be taken into consideration at the architecture design level, especially with a special emphasis on data minimization and purpose limitation principles.

**e) Regular Audits:** Compliance audits must be performed regularly and the cloud provider's compliance reports (SOC 2, ISO 27001) should be utilized to prove compliance.

ISSN: 2583-6129
DOI: 10.55041/ISJEM00113

**Identity and Access Management**

Challenge:

Sometimes it becomes challenging to safeguard data and follow the regulations including the GDPR, HIPAA, and PCI-DSS because of the decentralized cloud system.

Mitigation Strategies:

**a) Data Classification:** The data classification should also be developed to enable the safeguarding of the information of concern.

**b) Encryption:** Use better encryption techniques for the data that are stored as well as the data that are being transmitted. Leverage the cloud provider's key management services, or use bring your own key (BYOK) solutions.

**c) Privacy by Design:** Identify the need for privacy considerations in the architectural design and by so doing, adhere to the data minimization and purpose limitation principles.

**d) Regular Audits:** Perform compliance audits and align with the cloud provider's compliance reports like SOC 2 & ISO 27001 to show compliance.

**Threat Detection and Prevention**

Challenge:

Security threats' identification and immediate reaction in the case of distributed N-Tier architecture may be difficult because of a large amount of data and multiple interactions [10].


**Figure 4: SIEM**

(Source: https://cdn.prod.website-files.com/)

Mitigation Strategies:

**a) Security Information and Event Management (SIEM):** Integrate a cloud-native SIEM solution that would allow for the collecting and analyzing of security events occurring on all tiers.

**b) Intrusion Detection and Prevention Systems (IDS/IPS):** Employ Network and host-based IDS/IPS for the identification and mitigation of malicious activities.

**c) Behavioral Analytics:** Employ machine learning-based behavioral analytics to identify the anomalies and threats.

**d) Automated Response:** Implement automated incident response procedures to quickly contain and mitigate threats [11].

**e) Regular Penetration Testing:** Ensure penetration testing and vulnerability assessment are done severally to ensure all security loopholes are well captured.

**Encryption and Key Management**

Challenge:

The management of the encryption on different tiers and services and also the management of the keys can at times be complex and sometimes inaccurate.

Mitigation Strategies:

**a) Encryption Standards:** Limit access to information and implement strict data security policies and protocols, such as using AES-256 encryption on data at rest and data in transit at all tiers.

**b) Key Management Service (KMS):** Use the cloud provider's KMS, or engage a third-party KMS for key management.

**c) Key Rotation:** Ensure that key holders observe the key turnover policies that are recommended to reduce the effect of potentially compromised keys [12].

**d) Hardware Security Modules (HSMs):** For highly sensitive operations, consider using HSMs for key storage and cryptographic operations.

**e) Certificate Management**: Implement automated certificate management to prevent expired certificates and ensure proper validation.

**Disaster Recovery and Business Continuity**

Challenge:
Implementing business continuity and high availability for disasters or major outages in a distributed N-Tier architecture is challenging.
Mitigation Strategies:
**a) Multi-Region Deployment:** To achieve high availability and implement disaster recovery capabilities.
**b) Regular Backups:** Perform regular, encrypted backups of all critical data and configurations.
**c) Automated Failover:** Implement automated failover mechanisms to minimize downtime during outages.
**d) Disaster Recovery Plan:** Conduct a business impact analysis and come up with an efficient disaster recovery plan that should be updated frequently [13].
**e) Business Continuity Simulation:** Perform general business continuity activities to check on the level of readiness for different disasters.

## V. Best Practices for Implementing Secure N-Tier Architecture

It is important to have outstanding ideas about the design and implementation of a secure N-Tier architecture in the cloud environment.

### Design Principles

**a) Defense in Depth:**
Ensure that security controls are applied at different layers in the architecture. This approach helps the layers of protection where one layer fails to serve its purpose the other layers are still available to protect the system.
**b) Principle of Least Privilege:**
Give as few rights to the users, processes, and services as possible. This reduces the effects of security breaches to the minimum.
**c) Separation of Concerns:**
Ensure that the roles of each tier are well defined and that there is a clear line between each. This makes the system more secure, manageable, and easier to expand as the organizational or business needs increase.
**d) Secure by Default:**
Make security one of the most important structural components that should not be integrated at the last minute [14]. It is recommended that all the components should be secure in their default settings.
**e) Assume Breach Mentality:**
Design the architecture with the foresight that breaches are going to happen. Put in place efficient detection, response, and recovery measures.
**f) Immutable Infrastructure:**
Adopt immutability practices in infrastructure to maintain a healthy degree of uniformity as much as possible to minimize the attack vectors. It is more effective to replace the existing resources with new properly configured instances instead of updating them.
**g) Continuous Monitoring and Improvement:**
Conduct a continuous monitoring of all tiers and changes in security measures must be reviewed frequently due to the constantly emerging threats.

### Implementation Guidelines

**a) Network Security:**
Configure the network by using the Virtual Private Clouds (VPCs) and subnets.
Employ NACLs and security groups to regulate traffic flow in different tiers of architecture.
Perform Web Application Firewalls (WAF) to counter generic web risks.
**b) Data Security:**
Encrypt the data using suitable encryption algorithms when the data is idle (AES 256).
Always use the cloud provider's key management services for proper key management.
Employ DLP to ensure that there is restricted data leakage from the organization.
**c) Application Security:**
Adopt security principles and follow the best practices when coding and review the code often [15].
Concerning third-party library usage, it is required to utilize dependency scanning tools to address potential security risks.

Use input validation and output encoding to prevent injection attacks on the application.

d) Identity and Access Management:

Employ MFA for all the user accounts but especially for administrative ones.

Enforce the use of RBAC and conduct periodic reviews and rotation of access rights.

For administrative privileges on the other hand, one should adopt Just-In-Time (JIT) access techniques to reduce exposure to threats as much as possible.

e) Logging and Monitoring:

Ensure that there is logging for all the tiers of the architecture.

Implement the Security Information and Event Management (SIEM) for threat intelligence in real time.

It is the best practice to create alarms for these operations or any other type of suspicious activity and put into practice automated response protocols.

**f) Compliance and Governance:**

Another way of implementing security in the CI/CD pipeline is to integrate compliance audits on the deploys to conform to the set compliance standards.

Plan security audits and vulnerability scans and do them regularly.

## VI. Case Studies and Performance Analysis

This section provides real-life case studies of N-Tier architectures that are secure for computing in the cloud and discusses their efficiency and security aspects [16].

### Real-world Implementation Examples

### Case Study 1: Financial Services Company

A major multinational financial services company used a safe N-Tier architecture on Amazon to host its Internet banking.

Key components:

Presentation Tier: Elastic Load Balancer with Web Application Firewall (WAF)

Application Tier: ASG of EC2 instances with containerized microservices

Data Tier: Amazon RDS with Multi-AZ deployment and encryption at rest

**Security measures:**

Amazon VPC with Public and Private Subnets

AWS Shield for the protection against Distributed Denial of Service

AWS KMS that is suitable for the management of keys

CloudTrail in AWS for auditing and monitoring

Outcome: There was a report on the decrease in security incidences and compliance with the financial rules and regulations.

### Case Study 2: Healthcare Provider

A healthcare provider used secure N-Tier architecture on Azure to deploy patient records and telemedicine services.

Key components:

**Presentation Tier:** Web Application Firewall along with the Azure Front Door

**Application Tier:** Microsoft Azure's Kubernetes Service (AKS) for the microservices of an application.

**Data Tier:** With geo-replication, the right database option for the scenario is Azure Cosmos DB.

**Security measures:**

The Azure Virtual Network with network segmentation

For identity management and user authentication service, Microsoft Azure Active Directory.

Azure Key Vault as secrets management

Azure Security Center as the threat protection solution

**Outcome:** The provider succeeds in attaining HIPAA compliance besides cutting down on the latency of data access and system reliability.

### Performance Metrics and Evaluation

To evaluate the effectiveness of secure N-Tier architectures, several key performance indicators (KPIs) should be considered:

Security Metrics:

Number of security incidents

Time to detect and respond to threats

ISSN: 2583-6129
DOI: 10.55041/ISJEM00113

Percentage of failed security audits
Performance Metrics:
Response time
Throughput (requests per second)
Error rate
Availability Metrics:
Uptime percentage
Mean Time Between Failures (MTBF)
Mean Time To Recovery (MTTR)
Compliance Metrics:
Number of compliance violations
Time to address compliance issues
Availability = (MTBF / (MTBF + MTTR)) * 100
Where:
MTBF = Mean Time Between Failures
MTTR = Mean Time To Recovery
For example, if MTBF = 720 hours and MTTR = 1 hour:
Availability = (720 / (720 + 1)) * 100 = 99.86%
These case studies and the performance metrics provided in this paper show that secure N-Tier architectures, which are deployed in the cloud, can provide a substantial improvement in security, performance, and compliance [17]. Therefore, the philosophy underlying successful implementation is a careful design of changes, the introduction of the best practices, and regular evaluation and improvement.

## VII. Future Trends and Research Directions

The future of secure N-Tier architectures in cloud environments is likely to be shaped by several emerging trends:

**Zero Trust Architecture:** From perimeter-based security to no trust, everything has to be verified.

**AI-Driven Security:** Using machine learning and artificial intelligence for highly accurate threat identification, automated response, and proactive prevention.

**Quantum-Safe Cryptography:** To guard against the threats that quantum computing will pose in the future, incorporate quantum-resistant encryption algorithms.

Edge Computing Integration: Adding new nodes of N-Tier architectures to the apex level to create better performance and a lesser amount of latency.

Serverless Security: Writing new models of security for serverless architectures in the context of N-Tier systems [18].

**DevSecOps Evolution:** The continued inclusion of security into the developmental life cycle and a stronger focus on security testing and deployment.

**Blockchain for Integrity:** Analyzing the possibilities of using blockchain for data integrity in different tiers.

Studies in these fields are going to be essential in devising the future secure N-Tier structures for cloud computing that are capable of addressing emerging issues.

## VIII. Conclusion

N-tier architectures are traditionally more secure for cloud solutions. When security measures are taken in all the tiers it reduces risk, and compliance and enhances the performance of the organization. The future security of the cloud environments will thus require constant evolution to counter new threats and incorporate new technologies.

## IX. Reference List

### Journal

[1] Alseelawi, N.S., Adnan, E.K., Hazim, H.T., Alrikabi, H. and Nasser, K., 2020. Design and implementation of an e-learning platform using N-TIER architecture.

[2] Iqbal, W., Erradi, A., Abdullah, M. and Mahmood, A., 2019. Predictive auto-scaling of multi-tier applications using performance varying cloud resources. IEEE Transactions on Cloud Computing, 10(1), pp.595-607.

[3] Jonas, E., Schleier-Smith, J., Sreekanti, V., Tsai, C.C., Khandelwal, A., Pu, Q., Shankar, V., Carreira, J., Krauth, K., Yadwadkar, N. and Gonzalez, J.E., 2019. Cloud programming

simplified: A berkeley view on serverless computing. arXiv preprint arXiv:1902.03383.
[4] Morales-Sandoval, M., De-La-Parra-Aguirre, R., Galeana-Zapién, H. and Galaviz-Mosqueda, A., 2021. A three-tier approach for Lightweight data security of body area networks in E-health applications. IEEE Access, 9, pp.146350-146365.
[5] Ferrag, M.A., Maglaras, L. and Derhab, A., 2019. Authentication and authorization for mobile IoT devices using biofeatures: Recent advances and future trends. Security and communication networks, 2019(1), p.5452870.
[6] Jurvanen, K.J., 2021. Using AWS Secrets Manager with Kubernetes.
[7] Gu, Z., Nazir, S., Hong, C. and Khan, S., 2020. Convolution Neural Network-Based Higher Accurate Intrusion Identification System for the Network Security and Communication. Security and Communication Networks, 2020(1), p.8830903.
[8] Mahmood, H., Mahmood, D., Shaheen, Q., Akhtar, R. and Changda, W., 2021. S-DPS: An SDN-Based DDoS Protection System for Smart Grids. Security and Communication Networks, 2021(1), p.6629098.
[9] Alseelawi, N.S., Adnan, E.K., Hazim, H.T., Alrikabi, H. and Nasser, K., 2020. Design and implementation of an e-learning platform using N-TIER architecture.
[10] Ullah, F., Naeem, H., Jabbar, S., Khalid, S., Latif, M.A., Al-Turjman, F. and Mostarda, L., 2019. Cyber security threats detection in internet of things using deep learning approach. IEEE access, 7, pp.124379-124389.
[11] Wang, Q., Zhang, S., Kanemasa, Y. and Pu, C., 2019. Mitigating tail response time of n-tier applications: The impact of asynchronous invocations. ACM Transactions on Internet Technology (TOIT), 19(3), pp.1-25.
[12] Fair-Wright, C., 2021. Computerized Maintenance Management Systems (CMMS): The Evolution of a Maintenance Management Program. In Web Based Energy Information and Control Systems (pp. 121-134). River Publishers.
[13] Matsubara, Y. and Yamori, K., 2021. Survey on post-disaster timelines following a large-scale disaster expected to occur in the near future for pre-disaster recovery planning. Journal of Integrated Disaster Risk Management, 11, pp.26-45.
[14] Metan, J. and Murthy, K.N., 2019. N-tier modelling of robust key management for secure data aggregation in wireless sensor network. International Journal of Electrical and Computer Engineering, 9(4), p.2682.
[15] Vuong, T.H., Thi, C.V.N. and Ha, Q.T., 2021. N-tier machine learning-based architecture for DDoS attack detection. In Intelligent Information and Database Systems: 13th Asian Conference, ACIIDS 2021, Phuket, Thailand, April 7–10, 2021, Proceedings 13 (pp. 375-385). Springer International Publishing.
[16] Zhang, S., Shan, H., Wang, Q., Liu, J., Yan, Q. and Wei, J., 2019, July. Tail amplification in n-tier systems: a study of transient cross-resource contention attacks. In 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS) (pp. 1527-1538). IEEE.
[17] Knezevic, J. and Akademy, M.I.R.C.E., Failure Mechanisms and Probability-The Myth of MTBF.
[18] Datta, P., Kumar, P., Morris, T., Grace, M., Rahmati, A. and Bates, A., 2020, April. Valve: Securing function workflows on serverless computing platforms. In Proceedings of The Web Conference 2020 (pp. 939-950).